Assignment 1: Introduction to Object-Oriented Programming

Objectives:

1. Understand the basic concepts of OOP, such as objects, classes, attributes, and methods.
2. Explore the benefits of OOP, including modularity, reusability, and maintainability.
3. Compare OOP with procedural programming.

Instructions:

1. **Creating a Class and Objects**

   a. Define a class named `Car` that has the following attributes:

      `brand` (string)

      `model` (string)

      `year` (integer)

      `mileage` (integer)

   b. Create an `__init__` method to initialize these attributes.

   c. Define a method called `display_info()` that prints the details of the car in a readable format.

   d. Create three different objects of the `Car` class (e.g., representing three different cars) and call the `display_info()` method for each.

2. **Methods and Object Behavior**

   a. Add a new method called `drive()` to the `Car` class that:

      Takes a parameter `distance` (integer).

      Adds `distance` to the `mileage` attribute.

   b. Update the `display_info()` method to include the updated mileage after the car is driven.

   c. Create an object of the `Car` class, call the `drive()` method with a distance of 100, and then display the updated information of the car.

3. **Benefits of OOP**

   a. Create a new class named `ElectricCar` that inherits from the `Car` class. Add the following attributes:

   `battery_capacity` (integer, in kWh)
   `charge_level` (integer, percentage)

   b. Define a method `charge()` that takes a parameter `amount` (integer) and adds this amount to `charge_level`, ensuring that the charge level does not exceed 100%.

   c. Write a brief explanation (4-5 sentences) on how the `ElectricCar` class demonstrates the principles of modularity, reusability, and maintainability.

4. **Comparison of OOP and Procedural Programming**

   a. Rewrite the `Car` functionality (from Task 1) using a procedural programming approach.

   b. Answer the following questions:

   What differences did you notice between the OOP and procedural implementations?

   How does OOP provide benefits like modularity, reusability, and maintainability compared to procedural programming?

Submission:

1. Submit your Python code in a `.py` file.
2. Write the explanation in Task 3c and the answers for Task 4b as comments in your Python code.

Example Output for Task 1 and 2

```
Car Information:
Brand: Toyota
Model: Corolla
Year: 2020
Mileage: 15000

After driving 100 km...
Car Information:
Brand: Toyota
Model: Corolla
Year: 2020
Mileage: 15100
```